CESED - CENTRO DE ENSINO SUPERIOR E DESENVOLVIMENTO

CURSO: SISTEMAS DE INFORMAÇÃO

PROFESSOR: JOSÉ ANDERSON RODRIGUES DE SOUZA

COMPONENTE CURRICULAR: CONECTAR BANCO DE DADOS COM P-O-O

LISTA DE EXERCÍCIO 01

(Persistência a Dados, Tipos de Dados, XML, JSON e JDBC)

Alunos:

David Mickael Chaves de Moraes

Iago José Ramos Borges

Marcksuel Lopes de Menezes Sousa

José Lucas Bringel Leite

CAMPINA GRANDE/PB 2023

1. O que você entende por Persistência de Dados?

R - A persistência dos dados tem como objetivo garantir que as informações serão armazenadas em um meio que possa ser recuperada de forma consistente.

2. Quais as diferenças entre objetos transientes e objetos persistentes? Explique.

R - Objetos transientes são objetos que existem apenas enquanto a aplicação que os criou continuar executando, com isso após o término da aplicação eles deixam de existir. Para tornar estes objetos persistentes deve-se adotar uma das seguintes estratégias: Persistência por Classe: estratégia mais simples, porém não flexível.

3. Qual a principal diferença entre Banco de Dados Relacional e Banco de Dados Orientada a Objetos?

R - A principal diferença é o modelo de dados, banco de dados relacional, usa tabelas para armazenar dados, em formato tabular. Banco de dados orientado a objetos armazena dados a objetos, semelhantes a programação orientada a objeto, com atributos e métodos.

4. Qual o objetivo do mapeamento objeto-relacional (ORM)?

R - O objetivo do mapeamento objeto relacional é facilitar a integração entres sistemas, permitindo desenvolvedores acessem e manipulem banco de dados.

5. Defina:

a) Dados estruturados;

R - Informações organizadas em um formato específico e previsível, seguindo um esquema ou estrutura definida. Exemplos incluem bancos de dados relacionais, documentos XML, JSON, arquivos CSV e HTML com marcadores semânticos. Essa organização facilita a manipulação, consulta e análise por sistemas de gerenciamento de banco de dados ou algoritmos automatizados.

b) Dados semiestrurados;

R - Dados semiestruturados são informações com algumas estruturas, mas não rigorosamente tabulas, frequentemente representadas em formatos como JSON ou XML.

c) Dados não estruturados;

Dados não estruturados são informações que não seguem um formato especifico ou organização predefinida, incluindo texto livre, imagens, áudios e vídeos, sendo assim menos organizados e mais desafiadores.

6. Qual o nome da biblioteca responsável pela extração/captura de dados disponíveis em arquivos HTML ou XML? Explique.

R - É a JSoup, ela permite analisar estruturas em HTML e XML, extrair informações e navegar pelo conteúdo de forma que facilita a extração de dados.

7. Os arquivos do tipo XML (eXtensible Markup Language) surgiram como forma de estruturação e troca de dados pela internet. Dentre suas principais característica preencha os seguintes questionamentos:

a) Sintaxe inicial na primeira linha do arquivo.xml

R - A primeira linha define a versão do XML e codificação dos caracteres.
<?xml version="1.0" encoding="UTF-8"?>
“Adeclaração indica que o XML está na versão 1.0 e utiliza a codificação de caracteres UTF-8. A versão e a codificação podem variar dependendo do arquivo XML em questão.”

B) Os dados são organizados em formato hierárquico ou tabular?

Os dados em arquivos XML são organizados em um formato hierárquico. Isso significa que os elementos são estruturados em uma árvore, onde há um elemento raiz que contém outros elementos como filhos, e estes podem conter outros elementos como seus próprios filhos, formando assim uma hierarquia.Ex:

<livro>
<titulo>Harry Potter – O Prisioneiro de Azkaban</titulo>
<autor>J. K. Rowling</autor>
<ano>1999</ano>
</livro>

c) Quais são as formas de representação de um documento XML.

R - As formas de representação de um documento XML são:

Formato Textual: Comum e legível por humanos e máquinas, facilitando edição e leitura.
Formato Binário: Menos comum, mas otimizado para eficiência de armazenamento e transmissão.

Formato DOM (Document Object Model): Representa o XML como uma árvore de objetos na memória para manipulação.

Formato SAX (Simple API for XML): Permite a leitura sequencial de um documento XML, sendo eficiente em termos de memória.

Formato JSON: Uma alternativa popular ao XML, com sintaxe mais compacta, adequada para troca de dados estruturado

8. Elabore um documentoxml sobre produtos disponíveis para venda em empresas do comércio eletrônico/móveis/imóveis/roupas, a partir das seguintes condições:

▪ O produto deve possuir 5 características;
▪ Cada produto deve ter um nome de identificação;
▪ No documento deverá ter pelo menos dois produtos preenchidos.

R –

```
<?xml version="1.0" encoding="UTF-8"?>
<produtos>
   <produto>
      <tipo>Roupas</tipo>
      <nome>Camiseta Casual</nome>
      <caracteristicas>
         <cor>Azul</cor>
         <tamanho>Médio</tamanho>
         <material>Algodão</material>
         <estampa>Estampado</estampa>
         <preco>29.99</preco>
      </caracteristicas>
   </produto>
   <produto>
      <tipo>Móveis</tipo>
      <nome>Sofá de Couro</nome>
      <caracteristicas>
         <cor>Marrom</cor>
         <material>Couro genuíno</material>
         <estilo>Moderno</estilo>
         <dimensoes>210cm x 90cm x 80cm</dimensoes>
         <preco>899.99</preco>
      </caracteristicas>
   </produto>
</produtos>
```

9. Defina o que é um documento JSON e quais suas principais características.

R - Um documento JSON é um formato de troca de dados leve, independente de linguagem e fácil de ler e escrever. Suas principais características incluem uma sintaxe simples e legível, independência de linguagem, suporte a diferentes tipos de dados, estrutura hierárquica, ampla utilização na web, facilidade de leitura e escrita, eficiência no consumo de espaço, facilidade de transformação em objetos em programação e a ausência de suporte a comentários diretamente no documento. É amplamente empregado para representar e transmitir informações estruturadas entre sistemas, especialmente em ambientes web.

10. O que significa o processo de serialização (JSON.stringify) e desserialização (JSON.parse) de documentos do tipo JSON?

R - A serialização é o processo de converter objetos ou dados em formato JSON, tornando-os em uma sequencia de caracteres legível. A desserialização é o processo inverso, convertendo uma sequencia de caracteres JSON de volta em objetos ou dados utilizados em um programa.

11. Faça um exemplo de documento JSON a partir de dados sobre serviços de vendas online.

• Utilize dados do tipo, string, inteiro, array e objetos.

R –

```
{
  "servicosOnline": [
    {
      "nome": "Loja de Eletrônicos",
      "tipo": "Eletrônicos",
      "anoDeFundacao": 2010,
      "clientes": [
        {
          "nome": "João",
          "idade": 32,
          "endereco": {
            "rua": "Rua A",
            "cidade": "Cidade X",
            "estado": "Estado Y"
          }
        },
        {
          "nome": "Maria",
          "idade": 28,
          "endereco": {
            "rua": "Rua B",
            "cidade": "Cidade Z",
            "estado": "Estado W"
          }
        }
      ]
    },
    {
      "nome": "Loja de Moda",
      "tipo": "Roupas",
      "anoDeFundacao": 2015,
      "clientes": [
        {
          "nome": "Carlos",
          "idade": 45,
          "endereco": {
```

```
          "rua": "Rua C",
          "cidade": "Cidade M",
          "estado": "Estado N"
        }
      },
      {
        "nome": "Ana",
        "idade": 30,
        "endereco": {
          "rua": "Rua D",
          "cidade": "Cidade P",
          "estado": "Estado Q"
        }
      }
    ]
  }
 ]
}
```

12. Quais são as principais diferenças entre documentos do tipo JSON e XML.

R -JSON usa a sintaxe mais simples com pares de chave-valor, enquanto o XML usa marcações verbosas com tags. O JSON é mais compacto e legível para humanos, geralmente usado em desenvolvimento web, comparando com XML, é mais extensível e historicamente usado em integração de sistemas e documentos complexos.

13. Para que serve utilizar JDBC com Sistemas de Gerenciamento de Banco de Dados.

R - JDBC é usado para permitir que aplicativos Java se conectem a sistemas de gerenciamento de banco de dados, como MySQL, Oracle e etc. Permitindo a comunicação e manipulação de dados entre o aplicativo e o banco de dados.

14. Quais são os principais componentes durante a implementação do JDBC? Explique.

Durante a implementação do JDBC, os principais componentes são:
* Driver JDBC: Fornece a interface de comunicação entre Java e o banco de dados específico.
- URL de Conexão: Especifica o local do banco de dados e os detalhes da conexão.
- DriverManager: Gerencia um conjunto de drivers de banco de dados e estabelece a conexão.
- Connection (Conexão): Representa uma sessão com o banco de dados.
- Statement (Instrução): Usado para executar consultas SQL no banco de dados.
- ResultSet: Mantém os resultados de uma consulta SQL.
- SQLException: Trata erros relacionados ao banco de dados.

Gerenciamento de Transações: Permite o tratamento de um conjunto de operações como uma única unidade. Esses componentes trabalham juntos para facilitar a interação entre uma aplicação Java e um Sistema de Gerenciamento de Banco de Dados.

15. Cites restrições sobre a utilização do JDBC para sistemas atuais.

Ao utilizar o JDBC em sistemas atuais, é importante considerar:

1. Disponibilidade de Drivers: Certificar-se de que existem drivers JDBC compatíveis com o banco de dados utilizado.

2. Segurança contra Injeção de SQL: Usar PreparedStatements ou CallableStatements para prevenir ataques de injeção de SQL.

3. Compatibilidade de Versões: Verificar a compatibilidade entre a versão do JDBC e a do banco de dados.

4. Gerenciamento de Conexões: Abrir e fechar conexões de forma apropriada para evitar vazamentos de recursos.

5. Tratamento de Exceções: Lidar adequadamente com exceções SQLException para evitar interrupções inesperadas.

6. Desempenho: Considerar otimizações de consultas e transações para garantir boa performance.

7. Controle de Transações: Gerenciar transações corretamente para manter a integridade dos dados.

8. Prevenção contra Deadlocks: Projetar consultas e transações de forma eficiente para evitar deadlocks.

9. Escalabilidade: Planejar o código JDBC para lidar com um grande volume de operações de banco de dados.

10. Uso de Pools de Conexão: Considerar o uso de pools de conexão para melhorar a eficiência no gerenciamento de conexões.

11. Práticas de Codificação Segura: Implementar boas práticas de segurança para evitar vulnerabilidades.

12. Monitoramento e Log: Implementar monitoramento e logs para rastrear atividades relacionadas ao banco de dados.

Estas são considerações essenciais para assegurar o correto funcionamento e segurança ao utilizar o JDBC em sistemas modernos.